JOINT

Journal of Informatics and Technology, Vol 1, No 2, Tahun 2012
http://ejournal-s1.undip.ac.id/index.php/joint

RANCANG BANGUN E-COMMERCE TEMPLATE UNTUK APLIKASI CONTENT AND MANAGEMENT ELECTRONIC MALL (CAME-MALL)

Kamal Maulana T[1], Aris Sugiharto, M.Kom[2], Helmie Arif Wibawa S.Si, M.Cs[3]
Program Studi Teknik Informatika Jurusan Matematika FSM UNDIP Semarang

Abstrak: Kebanyakan e-commerce di Indonesia masih berorientasi ke pasar regional. Sehingga e-commerce tersebut masih menggunakan bahasa Indonesia. Adanya e-commerce template memudahkan pembuatan sebuah e-commerce. Namun, kebanyakan e-commerce template menggunakan bahasa Inggris sebagai default. Dibutuhkan aplikasi pembuat e-commerce yang menggunakan bahasa Indonesia dan memiliki fungsi-fungsi yang hanya dibutuhkan oleh pengguna pemula. Came-mall adalah salah satu aplikasi e-commerce template yang menggunakan bahasa Indonesia dan mengadopsi pembayaran transaksi berupa transfer antar bank. Came-mall dibangun dengan menggunakan model Unified Process. Came-mall digunakan untuk membuat e-commerce dan melakukan manajemen pengguna, tampilan, transaksi, serta testimoni dari pelanggan.

Kata kunci: E-commerce, template, Unified Process

Abstract: Most e-commerce in Indonesia still oriented in regional target. So, e-commerce in Indonesia using Indonesian. E-commerce templates makes e-commerce easier to build. Most circulating e-commerce templates using English as the default. E-commerce templates which uses only the general functions and uses Indonesian language is needed for beginner user. Came-mall is e-commerce template that uses Indonesian language and adopting transaction using bank transfer. Came-mall built with Unified Process model. Came-mall can built e-commerce and manage the users, theme, transaction, and customer's testimonial.

Keywords: E-commerce, template, Unified Process

1. PENDAHULUAN

E-commerce di Indonesia telah berkembang dengan hadirnya banyak *e-commerce*, misal Bhinneka.com, Tokobagus.com, dan lainnya. Namun, kebanyakan *e-commerce* tersebut masih berorientasi ke pasar regional. Sehingga, banyak *e-commerce* masih menggunakan bahasa Indonesia. Pembayaran transaksi *e-commerce* juga masih banyak yang menggunakan sistem transfer antar bank. Dengan adanya *commerce template* berbasis bahasa Indonesia akan membantu pengguna yang masih berorientasi ke pasar regional untuk meningkatkan produktifitas. Sejarah transaksi yang telah selesai dilakukan oleh pengguna sering langsung dihapus. Namun, dengan adanya arsip transaksi tersebut dapat diketahui jumlah penjualan yang telah dilakukan dan dapat menghasilkan laporan penjualan, serta dapat mengingatkan pembeli bahwa pernah dilakukan transaksi tersebut.

2. DASAR TEORI

2.1. E-commerce

[1] Mahasiswa Teknik Informatika UNDIP
[2] Dosen Pembimbing 1
[3] Dosen Pembimbing 2

E-commerce adalah segala bentuk transaksi yang menggunakan media elektronik, misalnya Internet [9]. Jenis *e-commerce* dibagi menjadi beberapa kategori. Pembagian jenis *e-commerce* berdasarkan pelaku (*actor*) yang terlibat. Ada yang membagi menjadi empat, lima[9] hingga enam[4] jenis. Dari beberapa pembagian *e-commerce* tersebut secara garis besar, *E-commerce* dapat dibagi menjadi dua yaitu:

1. *Business to Business* (B2B)
   B2B memiliki karakteristik sebagai berikut :
   a. Informasi yang terjadi hanya pada *partner* yang sudah diketahui dan umumnya memiliki hubungan kerja yang cukup lama. Sehingga informasi yang dikirimkan sesuai dengan kebutuhan dan kepercayaan
   b. Salah satu pelakunya dapat inisiatif mengirimkan informasi, tidak harus menunggu rekannya
   c. Model yang digunakan umumnya *peer-to-peer* (P2P). Proses dapat didistribusikan oleh kedua pelaku bisnis.
2. *Business to Customer* (B2C)
   Sedangkan B2C memiliki karakteristik sebagai berikut :
   a. Informasi yang diberikan bersifat umum dan dapat digunakan untuk khalayak ramai
   b. Pelayanan yang diberikan berdasarkan permohonan, bila konsumen melakukan permohonan, maka produsen baru memberikan respon sesuai permohonan
   c. Model yang digunakan adalah pendekatan *client/server*. Konsumen sebagai *client*, dan produsen sebagai *server*.

2.2. Basis Data

Basis data atau *database* adalah suatu susunan/kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu [6].

2.3. Database Management System (DBMS)

Yang dimaksud dengan DBMS adalah kumpulan file basis data yang saling berkaitan serta aplikasi untuk mengelolanya. Dapat dimisalkan, basis data adalah kumpulan datanya, sedangkan aplikasinya berdiri sendiri dalam suatu paket program untuk mengelola data yang memiliki suatu bahasa tertentu. Bahasa-bahasa dalam DBMS dibagi menjadi tiga, yaitu:

1. *Data Definition Language* (DDL)
   Berguna untuk menentukan struktur, membuat, serta menghapus *database*.
2. *Data Manipulation Language* (DML)
   Berguna untuk menentukan, dan memelihara isi dari *database*. Secara dasar terdapat dua tipe DML, yaitu:
   a. Prosedural
   User harus menentukan data apa yang dibutuhkan dan bagaimana mendapatkannya. Contoh: dbase III, foxbase.
   b. *Non-Prosedural*
   User harus menentukan data apa yang dibutuhkan tanpa menentukan bagaimana mendapatkannya. Contoh: SQL.
3. *Data Control Language* (*Query Language*)
   Menentukan akses terhadap *database*. Merupakan bagian DML yang akan digunakan untuk pengambilan informasi [6].

2.4. Content Management System(CMS)

CMS adalah aplikasi web yang membebaskan *user* untuk membuat, melakukan editing, dan mengorganisasi *content* website tanpa memprogram. CMS sangat banyak contohnya, misalnya: Joomla, AuraCMS, Wordpress. Kelebihan dari penggunaan CMS, adalah :

1. Mudah digunakan
2. Dapat digunakan siapa saja
3. Tidak perlu memikirkan proses yang terjadi di dalam CMS

2.5. Unified Model Language (UML)

UML adalah sebuah penggambaran model yang berfungsi untuk melakukan spesifikasi, visualisasi, membangun, dan dokumentasi dari sebuah *object-oriented software* yang sedang dibangun [5]. Dalam UML, untuk merepresentasikan sekumpulan elemen dan *relationship* dibutuhkan gambaran berupa diagram. Adapun beberapa diagram dalam UML, yaitu:

1. *Class Diagram*
   Class diagram menampilkan kumpulan *class*, *interface* dan *collaboration*, serta *relationship* yang merupakan gambaran umum desain sistem.

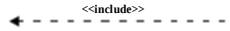
Tabel 1. Jenis *Relationship* pada *Class Diagram*

Jenis	Deskripsi	Gambar
Association	*Association* menghubungkan dua *class* dan menunjukkan hubungan semantik. *Association* dapat mempunyai arah yang menunjukkan *navigability*.	
Dependency	*Dependency* adalah relasi yang menunjukkan bahwa sebuah *class* menggunakan informasi atau layanan dari *class* lain.	
Generalization	*Generalization* menunjukkan hubungan generalisasi/spesialisasi.	
Aggregation	*Aggregation* menunjukkan hubungan "bagian dari" atau *whole/part hierarchy*.	
Composition	*Composition* menunjukkan jenis khusus dari *aggregation* dengan bagian bawah *multiplicity* selalu bernilai 1.	

2. *Use case Diagram*
   Use case diagram menunjukkan *actor*, dan *use case*, serta *relationship*-nya. *Use case diagram* sangat penting untuk pembuatan model dan pengorganisasian sistem.

Tabel 2. Jenis *Relationship* pada *use case diagram*

Jenis	Deskripsi	Gambar
Association	*Association* merupakan komunikasi antara aktor dan *use case* yang terlibat.	
Generalization	*Use case* anak mewarisi arti dari *use case* induk dan menambahkan/ memodifikasi *behaviour* dari induk.	
Extends	Menunjukkan ketergantungan suatu *use case* dengan *use case* yang lain, tetapi secara opsional	<<extends>>

Jenis	Deskripsi	Gambar
Include	Menunjukkan ketergantungan suatu *use case* dengan *use case* yang lain secara penuh.	

3. *Object Diagram*
   Object diagram menunjukkan kumpulan *object* dan *relationship*-nya.
4. *Sequence Diagram*
   Sequence diagram menunjukkan interaksi komunikasi pesan tiap *object*.
5. *Collaboration Diagram*
   Collaboration diagram menunjukkan struktur organisasi *object* yang menerima dan mendapatkan pesan.
6. *Statechart Diagram*
   Statechart diagram menunjukkan siklus hidup *object*, mulai dari *object* dihidupkan hingga *object* mati. Pada siklus ini juga menunjukkan perubahan *object*.
7. *Activity Diagram*
   Activity diagram menunjukkan *control* / *data flow* dari satu proses ke proses lainnya mulai dari awal hingga akhir sistem.
8. *Component Diagram*
   Diagram ini menunjukkan pengorganisasian dan ketergantungan *component* dalam sistem.
9. *Deployment Diagram*
   Menjelaskan mengenai bagian-bagian sistem yang akan dipergunakan nantinya, mulai dari *hardware*, *software*, *server*, serta perangkat lainnya.

2.6. Unified Process (UP)

Unified process adalah suatu metode penggunaan *use case*, arsitektur sentral, *iterative*, dan *incremental* dari pengembangan proses *framework* yang mengatur proses, orang, dan produk dari desain proses. UP memiliki ciri sebagai berikut [2]:

1. *Use case driven*
   Use case driven berarti menggunakan use case untuk gambaran kebutuhan utama dari sebuah sistem [2].
2. *Architecture-centric*
   Architecture-centric adalah suatu arsitektur yang digunakan agar dapat menggabungkan fungsionalitas dan *interface* yang ada [2].
3. *Iterative and incremental*
   Dari segi tata bahasa, *iterative* adalah perulangan, sedangkan *incremental* adalah berurutan. Biasanya proses dipecah-pecah menjadi beberapa bagian yang lebih kecil. Tiap bagian terdiri atas iterasi. Tiap iterasi dapat mencakup semua *workflow* dalam proses [5].

Selain itu UP juga memiliki siklus yang disebut *Life Cycle Phase* [5]. Siklus ini terbagi atas empat fase yaitu :

1. *Inception*
   Dalam tahap ini tujuannya adalah menerangkan ide, visi, dan cakupan dari sistem. Hasil dari *inception* adalah ide-ide.
2. *Elaboration*
   Pada tahap ini menjelaskan detail sistem yang akan dibangun, seperti arsitektur dan kebutuhan fungsional sistem. Hasil dari *elaboration* adalah arsitektur dasar.
3. *Construction*
   Tahap ini memfokuskan pada dalam penyelesaian analisa sistem, pelaksanaan sebagian besar *design* dan implementasi. Hasilnya sering disebut beta program.

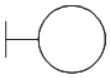
4. *Transition*
   Tahap ini melakukan penerapan sistem untuk digunakan di lingkungan *user*. Hasil dari tahap ini adalah produk akhir.

2.7. UP Workflow

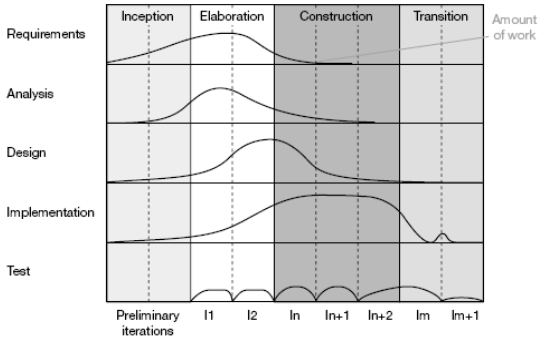
UP *Workflow* biasa disebut *Disciplines*. *Discipline* dilakukan pada tiap iterasi, dan tiap iterasi sendiri terorganisasi oleh *phases*. *Workflow* atau disiplin kerja dibagi menjadi lima yaitu:

1. *Requirement* (Kebutuhan)
   Requirement adalah fungsi yang akan dilakukan oleh sistem. Fungsi yang dibutuhkan dibagi menjadi dua yaitu *functional requirement* dan non-*functional requirement*. *Functional requirement* adalah kebutuhan yang ditawarkan oleh sistem. *Non-functional requirement* adalah sebuah *constraint* yang spesifik di dalam sistem.
   Hasil dari tahap ini adalah sebuah model *use case* yang terdiri atas *use case, actor* dan struktur *use-case* [5]. Pada penerapan UML akan digunakan diagram *use case*.
2. *Analysis*
   Disiplin ini memperbaiki dan membuat struktur dari *requirement* yang telah dibuat. Penjelasan tipe *analysis class* yang digunakan dapat dilihat pada tabel 3 [5]:

Tabel 3. Tipe *Analysis Class*

Jenis	Deskripsi	Gambar
Entity class	Menunjukkan data yang sudah ada dalam waktu lama.	
Boundary class	Menggambarkan interaksi antara sistem dan aktor. Interaksi meliputi masukan maupun keluaran.	
Control class	Menunjukkan fungsionalitas yang digunakan untuk mengatur interaksi antar *boundary class* dan *entity class.*	

3. *Design* (Model)
   Disiplin ini melakukan realisasi detail dari *requirement* ke dalam bentuk arsitektur sistem. Hasil yang akan dicapai adalah desain model yang meliputi *packages*, kelas, dan *interface*, serta asosiasinya dalam sistem. Selain itu, juga dihasilkan model penyebaran (*deployment model*). *Deployment model* mendeskripsikan bagaimana sistem akan disebarkan dalam kaitannya dengan komponen dan subsistem [5].
4. *Implementation* (Implementasi)
   Disiplin ini melakukan pembuatan sistem, mulai dari *coding*, hingga dokumentasi. Dalam tahap ini analisa dan desain diterapkan pada bahasa pemrograman. Selain itu juga dilakukan kompilasi, dan juga dokumentasi hingga menjadi sebuah program *beta*.
5. *Test* (Uji Coba)
   Disiplin ini memastikan apakah perangkat lunak sesuai dengan kebutuhan *user*.



Gambar 2.1 Hubungan Fase UP dengan UP Workflow dalam *Unified Process*

Pada gambar 2.1 menunjukkan hubungan antara ciri dan *workflow* pada UP. Bagian horizontal atas menunjukkan *Life Cycle Phase* UP. Sedangkan bagian horizontal bawah menunjukkan iterasi. Bagian vertikal menunjukkan UP *workflow*.

2.8. Black-box Testing

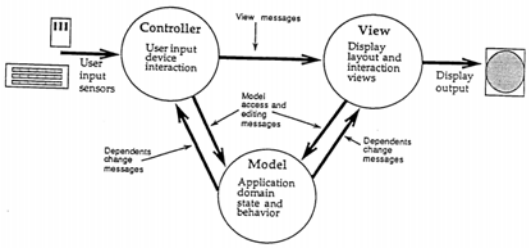
Black-box testing, atau *behavioral testing* berfokus pada functional requirement sebuah perangkat lunak [8]. Black-box testing dilakukan pada tahap akhir uji, karena black-box testing tidak memperhatikan struktur kontrol, tetapi berfokus pada domain informasi.

Tes ini bertujuan untuk:

1. Bagaimana validasi fungsional diuji ?
2. Bagaimana perfoma sistem diuji ?
3. Input apa saja yang akan membuat pengujian diterima ?
4. Apakah sistem sensitif terhadap masukan tertentu ?
5. Bagaimana batasan-batasan dari suatu data diisolasi ?
6. Berapa kecepatan data dan volume data yang ditoleransi ?
7. Apa pengaruh kombinasi tertentu dari data terhadap sistem operasi ?

2.9. Model-View-Controller(MVC)

Pemodelan MVC memisahkan pemodelan domain, presentasi, dan aksi berdasarkan masukan pengguna menjadi tiga bagian: *Model* (mengatur data dan memberi reaksi pada domain aplikasi), *View* (mengatur tampilan dari perangkat lunak), dan *Controller* (menterjemahkan masukan pengguna) [3].



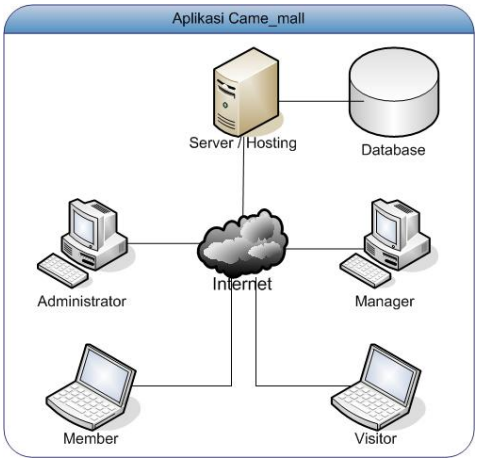
Gambar 2.2. Hubungan Model-View-Controller

3. Definisi Kebutuhan, Analisis, dan Perancangan

3.1. Definisi Kebutuhan (*Requirement*)

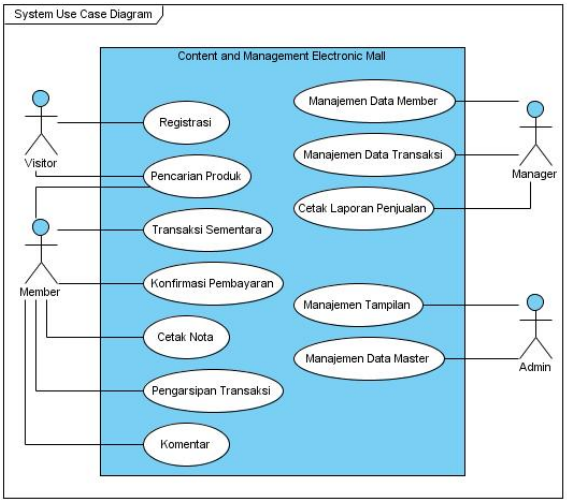
3.1.1. Deskripsi Umum Sistem

Sistem dibangun dengan menggunakan *software* yang menjadi standar di beberapa *Internet Service Provider* (ISP) di Indonesia, yaitu PHP dan MySQL.

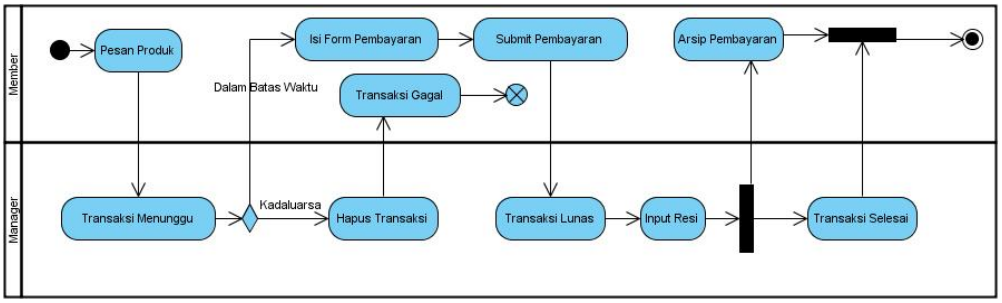


Gambar 3.1. Arsitektur Sistem Came-Mall

3.1.2. Model Bussiness



Gambar 3.2. Bussiness Use Case Diagram



Gambar 3.3. Activity Diagram Proses Transaksi pada Came-Mall

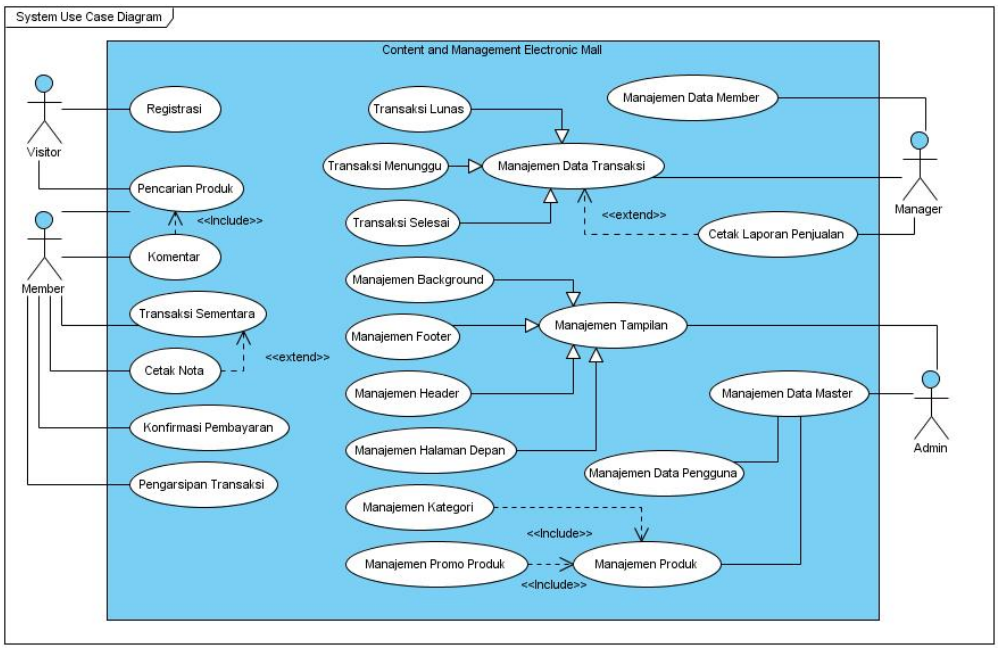
3.1.3. Daftar Actor

Tabel 4. Daftar Actor pada Came-Mall

No	*Actor*	Deskripsi
1	Admin	Pengelola yang bertanggung jawab penuh

2	Manajer	Pengelola yang menangani manajerial
3	Member	Calon pembeli yang terdaftar pada sistem
4	Visitor	Calon pembeli yang tidak terdaftar

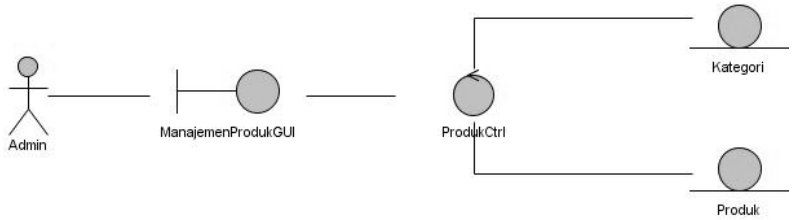
3.1.4. Sistem Use Case Diagram



Gambar 3.4. *System Use Case Diagram* pada Came-Mall

3.2. Analisa

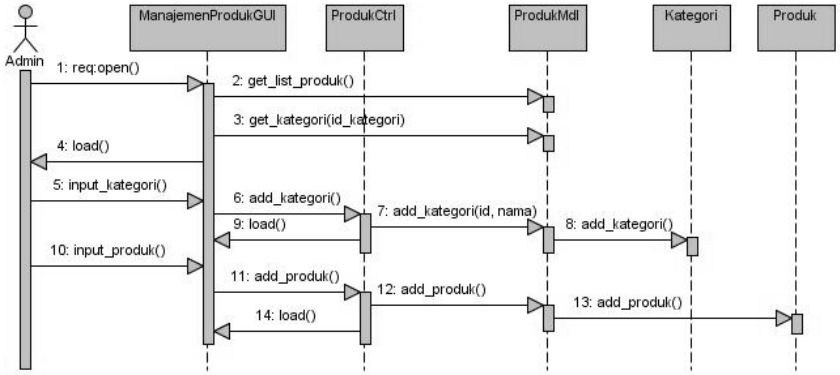
Dari *requirement* tersebut, dilakukan analisa dan mendapatkan hasil analisa dari pengembangan perangkat lunak berupa *use case*. Salah satu contoh realisasi *use case*, digunakan *use case* Manajemen Data Produk yang ditunjukkan *analysis class* pada gambar 3.5.



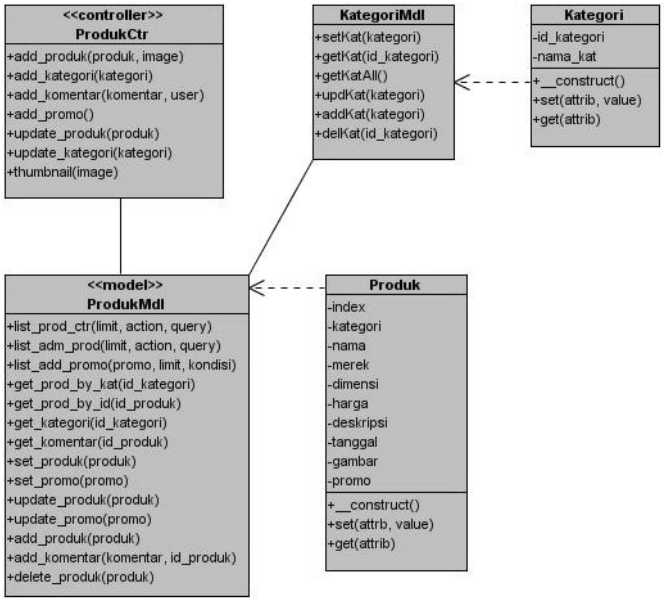
Gambar 3.5. *Analysis Class Model* Manajemen Data Produk

3.3. Perancangan

Realisasi *use case* merupakan kolaborasi objek perancangan yang berupa *use case*. Setiap *use case* direalisasikan dengan menggunakan *sequence diagram* dan *class diagram*.



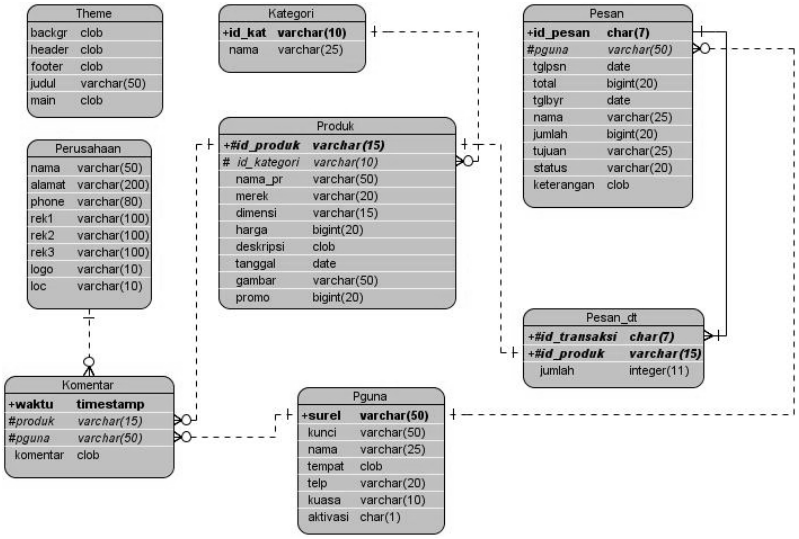
Gambar 3.6 *Sequence Diagram Use Case* Manajemen Produk



Gambar 3.7 *Class Diagram Use Case* Manajemen Produk

3.3.1. Perancangan Database

DBMS yang digunakan pada aplikasi came_all adalah basis data relasional. Rancangan skema basis data dapat dilihat pada gambar 3.8.



Gambar 3.8. Skema Basis Data pada Came_Mall

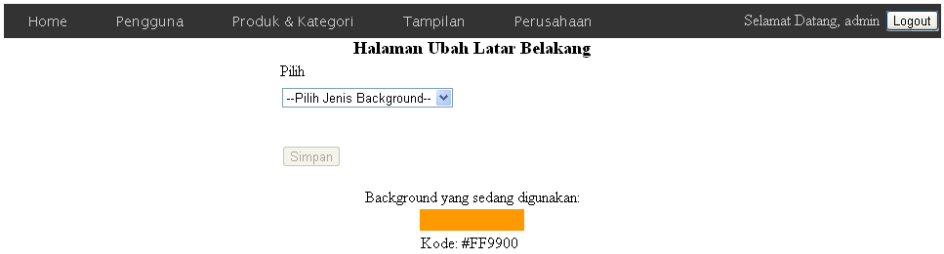
4. Implementasi dan Pengujian

4.1. Spesifikasi Implementasi

Implementasi dilakukan pada komputer dengan spesifikasi standar pada ISP dengan menggunakan XAMPP for Windows Version 1.7.2.

4.2. Implementasi Antarmuka

Digunakan contoh implementasi antarmuka pada halaman manajemen *background* ditunjukkan pada gambar 4.1.



Gambar 4.1. Implementasi Antarmuka Halaman Manajemen Background

4.3. Pengujian

Pengujian perangkat lunak dilakukan dengan metode *black-box*. Apabila fitur-fitur yang ada telah memenuhi *system requirements*, maka pengujian ini dikatakan berhasil.

5. Penutup

5.1. Kesimpulan

Pengembangan aplikasi Came-mall menggunakan UP dan menghasilkan e-commerce template yang:

1. Berbasis bahasa Indonesia
2. Menangani transaksi disertai nota pesanan.
3. Menangani manajemen produk, promo, *theme*, dan laporan pesanan.
4. Mengadopsi konfirmasi pembayaran melalui transfer bank
5. Menerima respon dari member.

5.2. Saran

1. Aplikasi ini dapat dibuat lebih menarik dengan mengganti bentuk tombol-tombol yang telah ada dengan gambar yang lebih mudah diingat.
2. Aplikasi ini dapat dibuat lebih interaktif bila digunakan proses yang berhubungan langsung dengan email.

6. Daftar Pustaka

[1] Arlow, Jim and Neustadt, Ila, 2002, "*UML and the Unified Process Practical Object-Oriented Analysis & Design*", Addison-Wesley.

[2] Booch Grady, Rumbaugh James, and Jacobson Ivar, 1998, "*The Unified Modeling Language User Guide*", Addison-Wesley.

[3] Burbeck, Steve, 1992, "*Applications Programming in Smalltalk-80(TM): How to use Model-View-Controller (MVC)*", ParcPlace Systems.

[4] F. Soesianto, Bressan Stephane, Ginanjar Herry, dan Ibrahim Ismail Khalil. "Mengembangkan Electronic Commerce di Indonesia: Aspek Teknologi, Bisnis, dan Hukum". Indonesian Information Society Initiative Gadjah Mada University, Yogyakarta.
[5] Hunt, John, 2003, "*Guide to the Unified Process featuring UML, Java and Design Patterns***",** Springer, London.
[6] Marlinda, Linda, 2004, "*Sistem Basis Data*", Penerbit Andi, Yogyakarta.
[7] Ojo, Adegboyega and Estevez, Elsa, 2005, "*Object-Oriented Analysis and Design with UML Training Course*", E-Macao.
[8] Pressman, R.S., 2001, "*Software Engineering : A Practitioner's Approach Fifth Edition*", McGraw-Hill, New York.
[9] Ruth, Zorayda Andam, 2003, "e-Commerce and e-Business", UNDP-APDIP